J. Symbolic Computation (1991) **11**, 275–289

Modelling Waves with Computer Algebra

D. A. DIVER

Department of Physics and Astronomy, University of Glasgow, Glasgow G12 8QQ, Scotland

(*Received* 9 *August* 1989)

A sophisticated model for linear waves in an inhomogeneous plasma is tackled completely using the computer algebra system REDUCE. The algebra code mirrors the mathematics, and is structured in a simple and straightforward manner. In so doing, the solution technique is made obvious, and the overall philosophy of the approach is intuitive to the non-specialist computer algebra user.

1. Introduction

In most areas of plasma physics, the mathematical models which describe the behaviour of the plasma (or ionised gas) are extremely complicated and minimally tractable, especially where the plasma equilibrium is non-uniform. Even in the simplest descriptions, permitting inhomogeneous background parameters leads to an almost impossible algebra burden which deters a systematic theoretical investigation.

In this paper, one such simple plasma model is presented fully worked as an example of the power and utility of computer algebra in such circumstances. The paper has the twin objectives of demonstrating how plasma theory can be developed by judicious use of computer algebra techniques, and how REDUCE may be used to mirror model mathematics in an intuitive manner, illuminating the method of solution, rather than the heavy mechanics of the manipulation itself. Hopefully, the latter purpose will appeal to those non-expert programmers like myself who often are dissuaded from using algebra packages by the apparent effort involved in extracting useful results from computer runs, and who are frustrated by the lack of simple examples in the literature.

The paper starts by defining the physical model, and lays out the analytical procedures required to progress to the desired solution. Then the corresponding REDUCE structure is detailed, with discussions on style and method. Finally, the concluding section contains output from running the codes in a REDUCE environment.

2. The Physical Model

A plasma is a fully ionised gas of electrons and ions, dominated by non-local forces resulting from the consequent electromagnetic interactions. In the simple cold plasma, the medium is treated as a perfectly conducting pressureless fluid which can support electromagnetic waves. A fuller description of basic plasma physics can be found in many texts, such as Stix (1962), or Boyd and Sanderson (1969).

Notwithstanding the physical interpretation of the model, the basic linearised equations governing the behaviour of the plasma fluid are

0747-7171/91/030275+15 $03.00/0

$$\frac{\partial n}{\partial t} + \nabla\cdot(n_0\mathbf{v} + n\,\mathbf{v}_0) = 0, \tag{1}$$

$$m\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}_0 + \mathbf{v}_0\cdot\mathbf{v}\right) = q\,(\mathbf{E} + \mathbf{v}_0\times\mathbf{B} + \mathbf{v}\times\mathbf{B}_0), \tag{2}$$

$$\nabla\times\mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}, \tag{3}$$

$$\nabla\times\mathbf{B} = \mu_0\left(\mathbf{J} + \varepsilon_0\frac{\partial \mathbf{E}}{\partial t}\right), \tag{4}$$

$$\mathbf{J} = n\,q\,\mathbf{v}, \tag{5}$$

where we have considered only one species of particle for simplicity. The symbols quoted have the following physical meanings: n is the number density of particles carrying charge q having mass m and moving with velocity $\mathbf{v}$; $\mathbf{E}$, $\mathbf{B}$ and $\mathbf{J}$ denote respectively the electric and magnetic fields in the plasma, and the plasma current. A subscript 0 on any of these quantities denotes the equilibrium value.

Following Diver *et al* (1990) we wish to study linearised perturbations about an inhomogeneous equilibrium, in which the magnetic field $\mathbf{B}_0$ has constant magnitude but varies in direction:

$$\mathbf{B}_0 = B_0\,[\hat{\mathbf{x}}\cos\phi(z) + \hat{\mathbf{y}}\sin\phi(z)]. \tag{6}$$

We will assume a constant rotation rate throughout, ie

$$\phi(z) = \phi' z, \quad \phi' = \textit{constant},$$

although this is a restriction which can be relaxed. Note also that we will take a periodic time dependence of all perturbed variables, so that

$$\partial/\partial t \rightarrow -i\,\omega$$

when operating on any first - order term. For physical reasons explained in Diver *et al* (1990), the equilibrium satisfies $\mathbf{v}_0\times\mathbf{B}_0 = 0$, $\mathbf{E}_0 = 0$, and $\mathbf{v}_0\cdot\nabla = 0$.

3. Solution Method

The goal of this analysis is to describe the kinds of waves it is possible to propagate in this non-uniform plasma. Thus we must derive a wave equation, and solve it, to get this information. Since there is only one independent variable (viz. z), we expect to derive an ordinary differential equation (ODE) and it is preferable to derive a homogeneous one (ie only one dependent variable throughout).

The plan of attack must be as follows.

(i) Solve the vector equation (2) for each perturbed velocity component in terms of each perturbed electric field component (remember, since this is a linear equation, this is always possible).

(ii) Then, using these relations, express the perturbed current $\mathbf{J}_1$ as a function of $\mathbf{E}_1$ through equation (5).

(iii) Taking the curl of equation (3), yielding

$$\nabla\times\nabla\times\mathbf{E} = \nabla(\nabla\cdot\mathbf{E}) - \nabla^2\mathbf{E} = -\partial/\partial t\,\nabla\times\mathbf{B}, \tag{7}$$

use equation (4) to substitute for $\nabla\times\mathbf{B}$ in terms of $\mathbf{J}$, so that in principle, equation (7) contains only the perturbed electric field $\mathbf{E}_1$ as the dependent variable. At this stage, it is still a vector equation. To derive a homogeneous ODE,

(iv) Eliminate two components in favour of the third.

The final step in the plan is

(v) Solve the resulting ODE.

As a scheme, this is not too difficult to define. However, the algebraic manipulation involved is very tedious, and extremely heavy. It is at this stage that the theorist may be tempted to make simplifications to ease the burden of the mathematics, thus compromising the generality and accuracy of any consequent analysis.

However, the next section illustrates how this solution plan may be implemented without regard to the overheads, by incorporating the structure in basic REDUCE commands.

4. Hardware and Software

Before discussing the construction of the REDUCE code, it is important to describe the computing environment used in this work. The work was performed on a High Level Hardware ORION 1/05, running OTS as an operating system (a direct port of UNIX 4.2), and the software was REDUCE 3.3, running Kyoto Common Lisp (KCL).

5. Constructing the REDUCE code

Having considered the best route to a solution, the next major task is to construct this path in REDUCE instructions.

5.1 notation

Since it is not possible to use identical notation in both the text of this document and the program listings, we must relate the code variables to the model notation. As a general rule, any variable name ending in x, y or z refers to the x,y or z component of that vector quantity. The equilibrium variables are identified in REDUCE by having the affix 0, with the perturbed, unknown quantities sharing the same names as in the text, but devoid of the subscript 1. Thus Ex is the x-component of $\mathbf{E}_1$, and so on.

5.2 assigning the variables

The first stage must be to define the basic parameters of the problem, such as the direction of the equilibrium fields, the functional dependence of the perturbed quantities, and so on. Thus we must examine the equilibrium, given by setting $\partial/\partial t$ of any variable to zero in the set (1)-(5). This allows us

to consider a time independent equilibrium, which is force-free if we also set $\mathbf{E}_0 = 0$, and then demand $\mathbf{v}_0 \times \mathbf{B}_0 = 0$. Note that $\mathbf{v}_0 \neq 0$, from (4), (5) and (6). These aspects of the model plasma are defined by the REDUCE program SETUP, quoted below.

In order to ease the notation, we have denoted the equilibrium velocity by **u**. Thus the magnitude of the equilibrium velocity is u_0, and that of the equilibrium magnetic field is B_0. All variables are perturbed quantities unless they possess the affix 0. Thus all the first-order variables have been made functions of z, by virtue of the Depend statements involving their x,y and z components.

5.3 coding the model equations

Having set up the basic problem, and assigned the appropriate variable dependencies, we must now tackle the full model differential equations. Clearly, (2) is the crucial equation to solve in order to express the perturbed velocity in terms of the perturbed electric field. Thus we must write the REDUCE equivalent of (2), component by component. This is done after the first Pause statement in program STAGE1, listed below. Notice that the advective derivative has been expanded fully, as has the vector cross-product.

Note that the perturbed magnetic field $\mathbf{B}_1$ appears on the right-hand side of (2) after linearisation. It can be eliminated in favour of $\mathbf{E}_1$ via equation (3). This is done, component by component, after the second Pause. Now REDUCE can substitute for $\mathbf{B}_1$ retrospectively, enabling the set eqn1, eqn2 and eqn3 to be solved for $\mathbf{v}_1$ in terms of $\mathbf{E}_1$, and constructing a list of the answers in velist. Thus the first stage in the solution technique has been achieved with minimal effort.

Now that we have $\mathbf{v}_1$ in terms of $\mathbf{E}_1$, we want to construct the perturbed current $\mathbf{J}_1$ using equation (5). However, when we linearise (5), we see that the perturbed number density n_1 enters the calculation for the first time. This must be eliminated in order to progress to a vector equation in $\mathbf{E}_1$ alone. This is achieved using the continuity equation (1), which when linearised yields n_1 as a function of $\mathbf{v}_1$. Note that this gives n_1 directly in terms of the components of $\mathbf{E}_1$, since the REDUCE environment can *already* eliminate $\mathbf{v}_1$ in favour of $\mathbf{E}_1$. Hence, we need do nothing other than write out equation (1) in REDUCE, and solve it for n_1. This is the first task undertaken by the program STAGE2.

Now we can proceed directly to expressing the perturbed current $\mathbf{J}_1$ in terms of $\mathbf{E}_1$ using equation (4) directly, and this is done after the first Pause in STAGE2, thus completing step (iii) in section 3.

The construction of the vector set of ODEs, stage(iv), involving only the components of $\mathbf{E}_1$ as the unknowns, depends on using equation (7). Again, it is sufficient to write out the components of (7) in order to effect the required substitutions and eliminations automatically. This is done after the second Pause in STAGE2.

The last instructions in STAGE2 allow the z component of $\mathbf{E}_1$ to be eliminated in the set of equations in favour of the other two components. This can be done directly, because no derivatives of E_{1_z} occurs in the z component of (7). Hence we now have two remaining coupled ODEs in two unknowns, E_{1_x} and E_{1_y}. During the course of the calculation, the two characteristic frequencies

$$\Omega = qB_0/m \quad (cyclotron\ frequency), \quad \omega_p = (n_0q^2/\varepsilon_0 m)^{1/2} \quad (plasma\ frequency)$$

appear in the working, and appropriate substitutions have been declared for them in the code. The presence of these quantities is very useful in the physical interpretation of the model structure.

5.4 solving the problem

So far we have used only the model equations to generate two coupled ODEs which govern the propagation of waves in our model plasma. However, we really need to know the detailed structure of the waves, and this can only be achieved by solving the ODEs. Thus for further progress we must advance from merely coding the actual model equations into developing a mathematical approach to solving the equations. However, the motivation behind the solution technique is partly based on physical grounds which may obscure the computational goal of this exercise, and so no detailed justification of the solution process will be provided.

As a general rule, when faced with coupled ODEs whose coefficients contain trigonometric functions of the independent variable, it may give extra insight if sin and cos are replaced by their exponential forms in the usual way:

$$\cos(x) = (e^{ix} + e^{-ix})/2, \quad \sin(x) = (e^{ix} - e^{-ix})/2i.$$

Although this may look more complicated, especially since the imaginary number i appears, it is actually a very useful transformation when combined with the construction of two new dependent variables, viz.

$$E_+ = E_{1_x} + iE_{1_y}, \quad E_- = E_{1_x} - iE_{1_y}.$$

The quantities E_+, E_- are important in analysing the polarisation of electromagnetic waves in a plasma. Hence the next appropriate step in solving our coupled equations is to transform the sines and cosines, and then add and subtract the two equations together in order to construct these polarisation variables in place of Ex and Ey.

This is done in the first part of STAGE3. On completion of this step, the resulting form of the equations reveals that a further simplification presents itself, namely the absorption of the common exponential factors in each equation through the further change of dependent variables from EPLUS and EMIN to

$$F_+ = E_+ e^{-i\phi'_2}, \quad F_- = E_- e^{i\phi'_2}.$$

The effect of all these transformations is to construct a pair of coupled ODEs with *constant* coefficients. This is the best possible result of the analysis, since such equations can be solved in a very simple manner.

However, there is a crucial point which must be stressed at this stage. The manipulations performed by STAGE3 are *not* obvious, nor can they be determined only from the context of constructing computer algebra code. In fact, they represent the experience of the mathematical modeller, and reflect an understanding of the physical (or mathematical) meaning of the model itself. The route to a solution is very often dictated by knowing the significance of the result. Clearly, computer algebra cannot provide this insight, but instead furnishes the modeller with the capacity to try out many ways of solving

the problem without the penalty of the accompanying algebraic manipulation. The final product is a refined algebra code which achieves the desired goal.

There remains the simple matter of deriving the dispersion relation itself. When an ODE has constant coefficients, such as

$$\frac{d^2y}{dx^2} + \Lambda y = 0$$

then the general solutions are of the form $y = Ae^{ikx}$ where $k = \sqrt{\Lambda}$. This is derived by substituting this form of solution into the differential equation, and solving the resultant algebraic relation for the unknown (k in this case). A similar procedure is used to find the nature of the general solutions to the coupled ODEs in the plasma model. Thus we substitute

$$F_+ = Ae^{(ikx)}, \quad F_- = Be^{(ikx)},$$

solve the first equation for B and thus eliminate it from the second. The resulting algebraic expression is then a fourth order polynomial in k: a dispersion relation. This polynomial then contains all the information required to characterise the wave solutions permitted in this medium.

Our problem is now completely solved. However, it is useful to rewrite the dispersion relation using a more compact and physically meaningful notation.

5.5 pattern searching

In the program STAGE4, the first half generates the polynomial in k, and the remaining code tidies up the output. In particular, we search the expression for factors using the gcd operator, which tests for the greatest common divisor of its arguments. Once these common factors are divided out, the expression can be further simplified by adopting a condensed notation, developed partly from uniform plasma dispersion relations. In our context, we wish to use the following definitions (Stix):

$$P = 1 - \sum_s \frac{\omega_{p_s}}{\omega}, \quad R = 1 - \sum_s \frac{\omega_{p_s}}{\omega^2} \frac{\omega}{\omega + \varepsilon_k \Omega_k}, \quad L = 1 - \sum_s \frac{\omega_{p_s}}{\omega^2} \frac{\omega}{\omega - \varepsilon_k \Omega_k}, \quad S = \tfrac{1}{2}(R + L),$$

where the summation is over all species s of plasma particle carrying a charge of $\varepsilon_s e$. In particular, when only considering 1 kind of particle, this reduces to

$$P = 1 - \frac{\omega_p^2}{\omega^2}, \quad RL/S = 1 + \frac{\omega_p^2(\omega_p^2 - \omega^2)}{\omega^2(\omega^2 - \Omega^2 - \omega_p^2)}. \tag{8}$$

The task is then to make the REDUCE code search the answer for these patterns and make the appropriate substitutions. For the casual user of computer algebra, this is usually the most frustrating aspect of using the system. Rarely are the patterns replaced in precisely the required manner, if indeed they are identified at all. Very often, a sequence of substitutions intended as simplifying steps tend to undo previous stages, and the inexperienced user then abandons the whole process, copying down the raw output on to paper and manipulating it by hand!

The golden rule in pattern searching and substitution is to avoid ambiguity in the forms declared as substitution candidates. For example, supposing our expression contains the quantities p and q and

we wish to declare a substitution for their sum and difference:

$$p + q = s, \quad p - q = d. \tag{9}$$

Mathematically, this is unambiguous and intuitively obvious. However, using (9) to construct a substitution in REDUCE will not work, since the two statements conflict: p is first declared to be $s-q$, and then subsequently $d+q$. The REDUCE code will use only the final substitution, and the candidate expression will not contain the variable s. This is because REDUCE only focuses on the first character string on the left-hand side of a substitution declaration, (generally speaking), and rewrites this quantity in terms of the remaining variables in the substitution declaration.

Clearly, an unambiguous method of implementing each substitution must be found. In this example, declaring

$$p = \tfrac{1}{2}(s + d), \quad q = \tfrac{1}{2}(s - d)$$

will yield the desired result. In fact, this technique has already been used in the substitutions for $\mathbf{E}_+$, $\mathbf{E}_-$ in STAGE2.

Returning to the dispersion relation, we wish to spot the occurrences of P and RL/S as given by (8). Thus we must find an unambiguous way of representing the candidate patterns. One such way is the following:

$$\varepsilon_0\mu_0(\omega^2 - \omega_p^2) = \frac{\omega^2}{c^2}P,$$

and

$$\varepsilon_0\mu_0\omega_p^2\Omega^2 = \frac{\omega^2}{c^2}\omega^2(\omega^2 - \Omega^2 - \omega_p^2)(P - RL/S)$$

This is one way of declaring the substitutions unambiguously (there are other ways, but this seems the most straightforward), and is implemented in the last half of STAGE4 (without the factor ω/c). The other substitutions involving λ_i are declared in a similar way, although the opportunity to redefine the expression is given after each substitution is declared. This is a useful tactic, in that it avoids undoing earlier forms of the expression as new ways of representing the terms are presented to the REDUCE environment. In almost every case, the definition of a useful substitution is entirely subjective. STAGE4 is merely an attempt to label gradient terms in a systematic way, and is not concerned with relations between them.

This completes the computer algebra code.

6. Concluding remarks

In the previous sections, a physical problem was defined in mathematical terms, and used as a template for constructing a REDUCE code. The problem was then solved completely by computer algebra, in a set of four short programs. These programs represent the distilled wisdom and experience gained from a process of continuous refinement of the solution method, the details of which were not reported here.

This is possibly the most important function of computer algebra in such a context. The mathematical modeller is able to sustain an analytic attack on a particular problem in a series of repeatable, verifiable and error-free steps, each building on the last, until a complete solution method is constructed in computer algebra. This piece of code can then be used to investigate all aspects of the problem, such as explicit parameter variations, which would normally require a completely separate attack. Of course, the pattern searching and final solution method may vary with the form of the parametrisation selected, but the underlying equations cannot change, and so the modeller may proceed confidently, knowing that at least the differential equations are correct. In most cases, it is the actual generation of the equations which is the main stumbling block in theoretical analysis.

The physical model given in this paper is not in any sense an artificially constructed exercise designed as a vehicle for the presentation of REDUCE. The end result is significant in the study of plasmas with sheared fields, and the REDUCE code has permitted subsequent development of the problem beyond the simple generation of a dispersion relation. However, the problem has touched on many aspects of computer algebra which may be useful for the novice (or the sceptic). The representation of differential equations, their manipulation and deployment of intermediate solutions, and the technique of pattern searching and factorisation have all been tackled as the demand arose. The programming is not sophisticated or particularly economical: the author is not an expert! Nevertheless, the object is to illustrate that the amateur can use simple and basic commands in computer algebra to achieve very powerful results.

7. Acknowledgments

It is a pleasure to acknowledge fruitful discussions on both the physical and computational aspects of this work with C C Sellar and E W Laing. The author is also grateful to the UKAEA Culham Laboratory for financial support.

8. References

Boyd, T.J.M., Sanderson, J.J. (1969). *Plasma Dynamics*. London: Nelson.

Diver, D.A., Laing, E.W. (1990). Mode Conversion and Transmission in Cyclic Fields. *Journal of Plasma Physics* **43**, 107-118

Diver, D.A., Laing, E.W. and Sellar, C.C. (1990). On waves in spatially cyclic equilibria. *Journal of Plasma Physics* **43**, 101-105

Kyoto Common Lisp (KCL), Research Institute for Mathematical Sciences (RIMS), Kyoto University, Kyoto, Japan.

REDUCE User's Manual Version 3.3 (1987), High Level Hardware Ltd, Oxford, England.

Stix, T.H. (1962) *The Theory of Plasma Waves*. New York: McGraw-Hill.

9. Appendix I: Program listings

```
%                          Program SETUP
COMMENT this file gives the equilibrium magnetic fields and velocities
        for the cold plasma model with rotating magnetic field. It is
        intended that this file is the setup routine for running stage1
        during a REDUCE session.

Pause;

Depend cosp, z$ Depend sinp, z$        % direction cosines function (z)

B0x := B0 * cosp $      B0y := B0 * sinp $      B0z := 0 $

ux := u0 * cosp $       uy := u0 * sinp $       uz := 0 $

Depend Ex,z$  Depend Ey,z$  Depend Ez,z$
Depend Bx,z$  Depend By,z$  Depend Bz,z$
Depend vx,z$  Depend vy,z$  Depend vz,z$

Df(cosp,z):=-sinp * dphi $    Df(sinp,z):= cosp * dphi $
    let cosp ^2 + sinp ^2 = 1 $

Comment at this stage, all eqm quantities should be defined, and all
        quantities should have the correct dependencies. Continue
        by calling up stage1.                                    $

end;
```

```
%                    Program STAGE1
Comment this is the first step in constructing the equations governing
        a cold plasma with a rotating magnetic field. First we write down
        the equations of motion governing a species s  and solve for the
        perturbed velocity vs in terms of the perturbed electric field E,
        with the eqm magnetic field B0 and eqm velocity us.

Pause;

eqn1 := -i*w*m*vx + m*vz*df(ux,z)
        - q*( Ex + vy*b0z - vz*b0y + uy*Bz - uz*By) $
eqn2 := -i*w*m*vy + m*vz*df(uy,z)
        - q*( Ey + vz*B0x - vx*B0z + uz*Bx - ux*Bz) $
eqn3 := -i*w*m*vz - q*( Ez + vx*B0y - vy*B0x + ux*By - uy*Bx) $

Pause;

Comment now solve for B in terms of E using dB/dt = - curl E;

Bx := ( df(Ez,y) - df(Ey,z) )/(i*w)$
By := ( df(Ex,z) - df(Ez,x) )/(i*w)$
Bz := ( df(Ey,x) - df(Ex,y) )/(i*w)$

Comment now solve for v in terms of E;

velist := Solve ( {eqn1,eqn2,eqn3},{vx,vy,vz} )$

Pause;

vx := rhs first first velist $
vy := rhs second first velist $
vz := rhs third first velist $

let q*B0/m = omc $              % defining the cyclotron freq $

Comment now we have v as a function of E and E'. Next we must calculate
        the current in terms of E and E'. To do this, we must take into
        account the number density perturbations.
      This ends stage 1. Continue by calling up stage2.   $
end$
```

```
%                    Program STAGE2
Comment this is the next stage in constructing the model equations for
        the cold plasma with rotating magnetic field. The perturbed
        current is expressed in terms of the perturbed E and E', taking
        account of number density perturbations.                    $

eqn4 := -i*w*n + df(n0*vz+n*uz,z)$              % continuity eqn
n := rhs first solve ( eqn4, n )$

Pause ;

Jx := q * ( n0 * vx + n * ux )$
Jy := q * ( n0 * vy + n * uy )$   % defining the currents $
Jz := q * ( n0 * vz + n * uz )$

Comment at this stage, we should be able to construct the dielectric
        tensor for the homogeneous model. Since this is not particularly
        useful in terms of the remaining calculations when performed by
        REDUCE, it is omitted.
      Instead go straight to the Maxwell equations curl E = - dB/dt,
        curl B = mu0 * (J + ep0 * dE/dt) :                          $

let n0*q^2/(ep0*m) = wp^2 $      % plasma frequency

Pause;

xcpt := df(Ex,z,2) + i*w*mu0*( Jx - i*w*ep0*Ex ) $
ycpt := df(Ey,z,2) + i*w*mu0*( Jy - i*w*ep0*Ey ) $
zcpt :=      i*w*mu0*( Jz - i*w*ep0*Ez ) $

%  zcpt defines Ez in terms of all the others ...

Ez := rhs first solve (zcpt, Ez) $

Comment now we are left with two equations involving only Ex and Ey $

Factor Df(Ex,z,2), Df(Ex,z), Ex, Df(Ey,z,2),Df(Ey,z),Ey $

Comment for further useful manipulation of these equations, load stage3 $
end;
```

```
%                        Program STAGE3
Comment in this part of the RMF suite of codes, the sines and cosines are
        converted into complex exponentials, E+ and E- are formed and then
      F+ and F-,yielding coupled ODEs with constant coefficients. $
Pause;

ss := xcpt + i*ycpt $ dd := xcpt - i*ycpt $

Comment now get rid of products of trigonometrics by substituting in favour
        of cos 2 phi and sin 2 phi:                                    $

let cosp^2 = (1+cos2p)/2, sinp^2=(1-cos2p)/2, cosp*sinp = sin2p/2;
Pause;
Comment next we form the E+ and E- variables in the usual way :                $
Depend eplus,z; Depend emin,z;
Factor df(eplus,z,2),df(eplus,z),eplus,df(emin,z,2),df(emin,z),emin;

let ex = ( eplus + emin )/2, ey = -i*(eplus - emin)/2;

%       express trigs in terms of complex exponentials ...

let cos2p = (e^(i*2*dphi*z)+e^(-i*2*dphi*z))/2,
   sin2p = -i*(e^(i*2*dphi*z)-e^(-i*2*dphi*z))/2;

On div;

Comment next form EPLUS = FPLUS e**(i*phi), EMIN = FMIN E**(-i*phi)
        in order to eliminate the z dependence in the coefficients.  $

Pause;

Depend fplus,z; Depend fmin,z;
Factor df(fplus,z,2),df(fplus,z),fplus,df(fmin,z,2),df(fmin,z),fmin;

Let Eplus = fplus*e^(i*dphi*z), emin = fmin * e^(-i*dphi*z);

comment the equations should now be ready for solution! Move to stage4 for
        final manipulation into a dispersion relation.                 $
;end$
```

```
%                    Program STAGE4
Comment here we sort out the dispersion relation itself, by substituting
        for FPLUS and FMIN, and calculating the secular determinant.    $

ss:=num (ss*e^(-i*dphi*z) )$      dd:=num ( dd*e^(i*dphi*z) )$

let FPLUS = AA*e^(i*kk*z),  FMIN = BB*e^(i*kk*z);
Pause;

l1:= solve ( dd/e^(i*kk*z),bb );
disprln := num ( sub( first l1, ss/(aa*e^(i*kk*z)) ) );
term := gcd ( coeffn(disprln,kk,4),coeffn(disprln,kk,2) );
dr := disprln / term;

Comment this is an attempt to match the patterns in all terms of the
        coupled odes for the rmf model. Note that to avoid back
        substitution, the pauses are included so that the candidates can
        be redefined as new vbles after each pattern matching.          $
Pause;

Let ep0*mu0*(w^2-wp^2) = PP;
Pause;

Let ep0*mu0*wp^2*omc^2/(omc^2+wp^2-w^2) = RLS - PP;

Pause;

Let ep0*mu0*u0^2*wp^2 = (w^2-wp^2-omc^2)*rlam1;

Pause;

Let ep0*mu0*u0*omc*wp^2 = (w^2-wp^2-omc^2)*rlam2;

Pause;

Let ep0*mu0*u0*omc*dphi = (w^2-wp^2-omc^2)*rlam3;

% ----->>  this is the end.....
```

10. Appendix II : selected output

In this section, output is quoted from particular stages in the running of the codes. The examples are chosen as concise illustrations of the development of the analysis at key points in the progress towards a solution.

For example, to show that STAGE1 really has solved for the velocities in terms of the electric fields, the response to the command

```
VSZ := (Q*(DF(EX,Z)*COSP*U0 + DF(EY,Z)*SINP*U0 + COSP*EY*OMC - EX*

                                                        2     2
      OMC*SINP + EZ*I*W))/(M*(DPHI*OMC*U0 - OMC  + W ))
```

showing that the required substitutions have been made. Similarly, if we ask for the expression defining the perturbed number density after the first Pause in STAGE2, the response is

```
N := (N0*Q*(DF(EX,Z,2)*COSP*U0 - DF(EX,Z)*DPHI*SINP*U0 - DF(EX,Z)

     *OMC*SINP + DF(EY,Z,2)*SINP*U0 + DF(EY,Z)*COSP*DPHI*

     U0 + DF(EY,Z)*COSP*OMC + DF(EZ,Z)*I*W - COSP*DPHI*EX*

     OMC - DPHI*EY*OMC*SINP))/(I*M*W*(DPHI*OMC*U0 -

     2     2
   OMC  + W ))
```

Obviously the status of any variable can be checked at all stages. However some of the expressions are quite lengthy, and since they are only steps on the way to the solution, they are not necessarily of interest to the reader, except that they give insight into the code structure. The next two examples show the status of the z components of the perturbed current and electric field at the end of STAGE2:

```
                2
JZ := (EP0*WP *(DF(EX,Z)*COSP*U0 + DF(EY,Z)*SINP*U0 + COSP*EY*OMC

                                                              2     2
                  - EX*OMC*SINP + EZ*I*W))/(DPHI*OMC*U0 - OMC  + W

          )

                                    2                     2
EZ :=  - (DF(EX,Z)*COSP*I*U0*WP  - EX*I*OMC*SINP*WP  + DF(EY,Z)*I*

                    2                     2
          SINP*U0*WP  + EY*COSP*I*OMC*WP )/(W*(DPHI*OMC*U0 -

              2    2     2
            OMC  + W  - WP ))
```

Finally, it is important to state the actual dispersion relation as calculated by REDUCE, since this is the desired result. In fact, the output shows the dispersion relation when common factors have been divided out:

```
          4                          2           2                2
DISP := KK *(RLAM1 + RLAM3 + 1) + KK *( - 2*DPHI *RLAM1 - 2*DPHI *

                        2
          RLAM3 - 2*DPHI  + 2*DPHI*RLAM2 - PPS*RLAM1 - 2*PPS*RLAM3 -

                       4             4             4          3
          PPS - RRS) + DPHI *RLAM1 + DPHI *RLAM3 + DPHI  - 2*DPHI *

                   2                    2                 2
        RLAM2 - DPHI *PPS*RLAM1 - 2*DPHI *PPS*RLAM3 - DPHI *PPS -

           2                              2
        DPHI *RRS + 2*DPHI*PPS*RLAM2 + PPS *RLAM3 + PPS*RRS
```

note that the wave number is denoted by KK. This result has been used in subsequent work in plasma physics (see Diver and Laing, 1990).